

# Classificação de Restrições de Integridade em Bancos de Dados Temporais de Versões

Robson Leonardo Ferreira Cordeiro, Clesio Saraiva dos Santos, Nina Edelweiss, Renata de Matos Galante

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

e-mail: [rlfcordeiro, clesio, nina, galante]@inf.ufrgs.br

## Resumo

*Este artigo propõe uma classificação de restrições de integridade para bases de dados com suporte a tempo e versões. Esta proposta tem o intuito de auxiliar trabalhos futuros de definição, especificação e verificação de restrições sobre esse tipo de base de dados, além de servir de base para a definição de procedimentos de otimização na manutenção da integridade dos dados. Restrições são detalhadamente analisadas, com base em seus aspectos origem, substância, especificação, aplicação, temporalidade e versionamento. Assim, a classificação pode ainda ser facilmente adaptada para sua utilização sobre bases de dados tradicionais ou que utilizem apenas um dos conceitos de tempo e versão. Além da definição das classes, exemplos comuns de restrições são classificados de acordo com suas características.*

## Abstract

*This paper proposes an integrity constraints classification for temporal versions databases. The main goal of this classification is helping future researches on their definition, specification and verification over this kind of database. Also, researches on the optimization of data integrity maintenance may be based on it. Constraints are deeply analyzed based on their origin, substance, specification, application, temporality and versioning aspects. Thus, the classification can also be easily adapted for snapshot databases or for the ones that use only one of the time and version concepts. Besides the definition of classes, common constraints samples are classified, based on their characteristics.*

## 1. Introdução

Um Sistema gerenciador de Bancos de Dados (SGBD) possui como principal característica a capacidade de gerenciar bases de dados que representam parte do mundo real. Para permitir uma representação fiel, os dados presentes na base devem obedecer a diversas restrições de integridade, que podem provir da aplicação modelada ou de outras fontes. Além disso, o suporte oferecido por um SGBD tradicional não é suficientemente adequado a aplicações com necessidades não convencionais [8, 12]. Muitas aplicações necessitam armazenar históricos de alterações e registrar períodos de validade de dados. Entre elas estão sistemas de controle financeiro, reserva em hotéis, companhias seguradoras, contabilidade e bancos de dados multimídia. Além disso, algumas aplicações necessitam armazenar versões de conjuntos de dados ou objetos gerenciando suas agregações e formas de representação. Entre elas estão ferramentas do tipo CAD (*Computer-Aided Design*), CASE (*Computer-Aided Software Engineering*) e de controle de produção de software. Através do suporte aos conceitos de tempo e versão, quando

provido pelo SGBD, grande parte dessas necessidades é suprida [8]. Assim, a utilização de bancos de dados com suporte a esses conceitos é cada vez mais necessária e comum.

Denomina-se Banco de Dados Temporal de Versões (BDTV), qualquer base de dados que possua uma estrutura adequada para suporte à utilização de tempo e versões simultaneamente. Basicamente, esse suporte pode possuir: (i) implementação integral na aplicação; (ii) implementação conjunta na aplicação e no SGBD; ou (iii) implementação integral no SGBD. Essa última é considerada a ideal, pois viabiliza o uso dos conceitos de tempo e versão, tornando a implementação o mais transparente possível no nível da aplicação.

Inúmeras pesquisas e implementações na manutenção da integridade de bancos de dados tradicionais (*snapshot databases*) foram efetuadas, mas restrições de integridade que considerem o gerenciamento de tempo e de versões representam uma área de pesquisa praticamente inexplorada.

Considerando essa realidade, o principal objetivo do presente trabalho é prover uma classificação completa de restrições de integridade para bancos de dados que suportem tempo e versões. Essa classificação auxiliará trabalhos futuros em sua definição, representação e verificação, além de servir de base para processos de otimização na manutenção da integridade dos dados.

Este artigo está estruturado da seguinte maneira: a segunda seção descreve brevemente o modelo TVM que é utilizado no estudo de caso apresentado e serviu de inspiração no desenvolvimento da classificação; a terceira e principal seção apresenta a classificação completa de restrições em qualquer BDTV, com base em diversos aspectos; na seção quatro é mostrado um estudo de caso sobre a classificação; na quinta seção são descritos os principais trabalhos relacionados; e a sexta e última seção apresenta conclusões e trabalhos futuros.

## 2. Modelo Temporal de Versões

O Modelo Temporal de Versões (TVM – *Temporal Versions Model*) [12] é um modelo de dados orientado a objetos que suporta uniformemente os conceitos de tempo e versão, considerando a existência conjunta de objetos tradicionais e objetos temporais de versões. Nesse modelo, o conceito de tempo é utilizado para controlar e armazenar o histórico de alterações sobre os dados da base, enquanto o conceito de versão permite gerenciar diversas alternativas de projeto.

O tempo pode ser associado a objetos, versões, atributos e relacionamentos. Um mesmo objeto apresenta uma linha de tempo para cada uma de suas versões. Assim, duas ordens temporais distintas são utilizadas, o tempo ramificado para um objeto e o tempo linear para suas versões. Além disso, o tempo varia de forma discreta e é representado através de rótulos de tempo intervalar, bitemporal (tempo de validade e tempo de transação) e implícito. Nesse modelo, o *template TV* é utilizado para representar classes temporais de versões, enquanto atributos e relacionamentos temporais são definidos através dos rótulos <<*T*>> e <<*Temporal*>> respectivamente. Um exemplo de modelagem no TVM pode ser visto através do diagrama de classes presente na Figura 9a.

É importante citar que a classificação de restrições de integridade apresentada neste trabalho se aplica a qualquer base de dados temporal de versões, independentemente do modelo de dados utilizado. Entretanto, julga-se necessária a existência desta seção, pois o modelo TVM foi utilizado como base do estudo de caso apresentado, além de ter sido a principal inspiração durante o desenvolvimento da classificação.

### 3. Classificação de Restrições de Integridade

Nesta seção, serão classificadas restrições de integridade em qualquer BDTV a fim de possibilitar uma análise profunda sobre suas diversas características. Segundo Santos [15], a especificação completa de uma restrição é composta pelos seguintes componentes: (i) *restringente*, que é um conjunto composto por objetos, armazenados ou não na base de dados, utilizados no estabelecimento de condições a serem respeitadas pelos dados do banco de dados; (ii) *restringido*, que corresponde ao conjunto de objetos, armazenados na base, cujos conteúdos são restringidos pela restrição; (iii) *condição restritiva (asserção)*, que consiste em uma expressão lógica que relaciona os integrantes do *restringente* e do *restringido*; (iv) *pontos de verificação*, que correspondem aos momentos de início do processo de verificação de validade da restrição; e (v) *ações de violação*, que correspondem às ações executadas sobre a base logo após a violação da restrição, com o propósito de recompor a integridade da base.

A definição do conjunto de *pontos de verificação* de uma restrição deve obrigatoriamente ser efetuada antes de sua ativação, exceto em restrições de *acionamento homogêneo manual*. A manutenção da integridade dos dados, em relação a uma restrição, é efetuada através da verificação da validade de sua *condição restritiva* em seus *pontos de verificação*.

Supõe-se neste trabalho que os objetos restringidos diretamente por uma restrição devem pertencer a um único domínio de esquemas, entidades, relacionamentos ou valor. Essa consideração também se faz válida quanto aos domínios de temporalidade e versionamento. Restrições com restringidos de domínios múltiplos não foram consideradas, pois elas sempre podem ser decompostas em duas ou mais restrições de domínios únicos [15]. Assim, critérios e sub-critérios relacionados diretamente ao restringido não classificam a homogeneidade, diferentemente dos relacionados ao restringente, que classificam a homogeneidade e sub-classificam restrições *homogêneas* de acordo com seus domínios. Restrições *heterogêneas* não são explicitamente sub-classificadas, mas poderiam ser, de acordo com as possíveis combinações das classes homogêneas.

O termo restrição de integridade, considerado neste trabalho, diz respeito não somente ao conceito pelo qual são restringidos os dados de uma base, pois diversos aspectos considerados dizem respeito à forma de aplicação dos conceitos e à forma com que a integridade da base é assegurada. Para tanto, alguns aspectos consideram também características relacionadas à definição e a especificação da restrição, além de detalhes do esquema, do SGBD e do modelo de dados adotado.

Com base em seus componentes e nas considerações apresentadas, restrições de integridade serão analisadas através dos seguintes aspectos: (i) *origem*, que diz respeito ao agente pelo qual elas se originaram; (ii) *substância*, que evidencia o tipo de condicionamento imposto aos dados; (iii) *especificação*, que considera sua forma de especificação; (iv) *aplicação*, que analisa as formas de manutenção da integridade dos dados em relação a elas; (v) *temporalidade*, que diz respeito às características temporais de seus restringidos e restringentes; e (vi) *versionamento*, que considera as características de versionamento dos mesmos conjuntos.

Objetivou-se classificar restrições de integridade em qualquer BDTV da forma mais ortogonal possível. Para isso, foram definidos aspectos pelos quais são analisadas as restrições. Cada aspecto possui critérios e sub-critérios a fim de possibilitar a definição de classes de restrições, preservando ao máximo sua independência, desde que não se afetasse a abrangência da classificação.

As seções seguintes descrevem, de acordo com a limitação de espaço existente, cada um dos aspectos citados. A classificação descrita em cada uma destas seções é apresentada graficamente através de árvores cujos nodos folha representam classes de restrições de integridade.

### 3.1. Origem

Restrições de integridade podem possuir diferentes origens (Figura 1). Considerando que o termo empresa diz respeito à aplicação representada na base, são estabelecidas as seguintes classes: (i) *ambiente*, que trata das restrições impostas por agentes externos à empresa, provenientes de fontes *naturais* ou *legais*; (ii) *empresa*, que engloba restrições impostas pela forma de funcionamento da empresa; (iii) *modelo de dados*, que engloba as restrições provenientes da inadequação do modelo à informação representada ou de um mau projeto de esquema; e (iv) *implementação*, que agrupa restrições que buscam adequar os dados às estruturas fisicamente utilizadas.

A quantidade e o detalhamento das restrições originadas pela *empresa* costumam ser proporcionais ao nível de automação e complexidade de sua implementação e verificação na base resultante [15].

De acordo com o foco principal deste trabalho, considera-se que restrições originadas no *modelo de dados* podem ser provenientes do controle de suas características de *temporalidade*, de *versionamento* ou de *outras de suas características*.

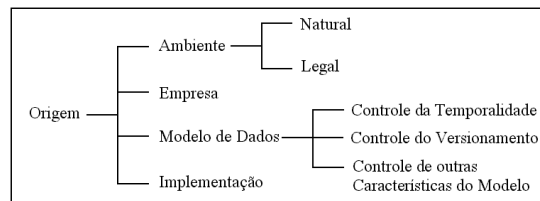


Figura 1. Classificação quanto ao aspecto origem.

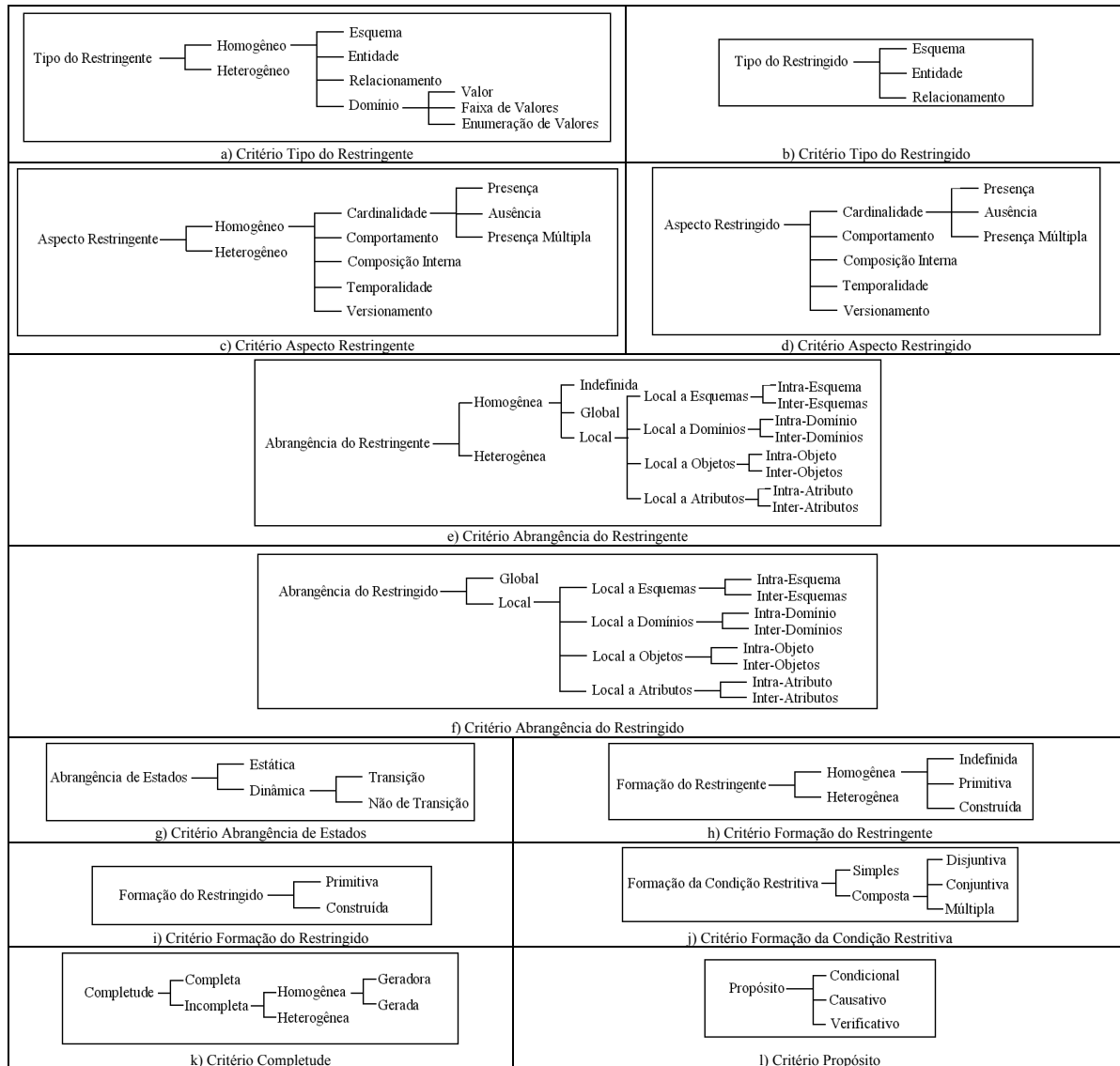
### 3.2. Substância

O aspecto *substância* permite a diferenciação de restrições com base em diversas características de seu significado. O principal ponto, aqui evidenciado, diz respeito ao tipo de condicionamento, imposto pela restrição, à base de dados. Este aspecto pode ser complementado pelos aspectos *temporalidade* e *versionamento*, criados a fim de agrupar ao máximo características temporais e de versões. A seguir serão expostos os critérios relacionados a este aspecto.

Quanto ao critério *tipo do restrigente* (Figura 2a), restrições podem ser *homogêneas* ou *heterogêneas*, de acordo com os tipos dos integrantes de seus restrigentes. Restrições *homogêneas* podem ter restrigentes de tipo *esquema*, *entidade*, *relacionamento* ou *domínio*, sendo que o último é dividido em *valor*, *faixa de valores* e *enumeração de valores*. Em relação ao *tipo do restringido* (Figura 2b), restrições são classificadas de forma análoga, exceto por não se considerar a homogeneidade e nem mesmo o tipo *domínio*. Esta última consideração é feita pois restringidos devem estar obrigatoriamente relacionados a componentes da base de dados.

Restrições de integridade podem ter validade dependente dos seguintes aspectos: *cardinalidade* ou *composição interna* dos dados, *comportamento* do esquema ou características de *temporalidade* ou *versionamento* do restrigente e restringido. A *cardinalidade* restringe os dados em relação ao número de objetos, presentes na base, com determinadas características comuns, inclusive características temporais e de versionamento. Ela pode se referir à existência de um único objeto (*presença*), à não

existência de determinado objeto (*ausência*) ou à existência de um determinado número (maior do que um) de objetos (*presença múltipla*). Considera-se também que restrições com restridentes ou restringidos de *tipo esquema* são as únicas capazes de utilizar o aspecto *comportamento*. Em um ambiente orientado a objetos, este aspecto corresponde a estabelecer diretivas de consistência a métodos, como restrições sobre domínios de parâmetros ou seqüências de execução [11]. Assim, as Figuras 2c e 2d ilustram o *aspecto restrigente e restringido*.



**Figura 2. Classificação quanto ao aspecto substância.**

Um dos principais critérios considerados durante a otimização da verificação de restrições é o escopo, ou seja, à área da base de dados analisada na sua verificação. A abrangência de uma restrição está intimamente ligada a sua complexidade e à quantidade de objetos examinados durante sua verificação. Uma restrição pode ter *abrangência do restrigente* (Figura 2.e): (i) *indefinida*, quando o mesmo for de *tipo domínio*, (ii) *global*, quando ele englobar toda a base de dados ou, (iii) *local*, quando ele englobar apenas partes específicas da base. A última classe é dividida em *abrangência local a esquemas*, que possui sentido quando aplicado à base o suporte à evolução de esquemas, *local a domínios*, *local a objetos* e *local a atributos*. Além disso, cada uma dessas classes é sub-classificada em *intra-classe* e *inter-classes*. A *abrangência do restringido* (Figura 2f) é classificada de

forma análoga, exceto pela impossibilidade de abrangência *heterogênea* e *indefinida*. Deve-se ainda ressaltar que a abrangência de restrições temporais de versões é complementada nos aspectos *temporalidade* e *versionamento*.

O critério *abrangência de estados* diz respeito aos estados da base de dados que devem ser conhecidos e utilizados durante o processo de verificação de uma restrição. De acordo com este critério, classificam-se restrições como *estáticas*, quando utilizam um só estado ou *dinâmicas*, caso contrário [1, 2, 3, 15]. Restrições *estáticas* restringem quais dados podem fazer parte da base de dados, enquanto restrições *dinâmicas* restringem a evolução desses dados [2]. Uma restrição *dinâmica* pode ainda ser de *transição*, quando utilizar dois estados que se sigam no fluxo do tempo, ou *não de transição*, caso contrário. Assim, a classificação pode ser vista na Figura 2g.

Divergindo de outros trabalhos [5, 9], esta classificação considera que restrições *estáticas* também podem ser *temporais*, pois elas podem referenciar um único ponto no tempo passado, presente ou futuro. Já restrições *dinâmicas*, são *obrigatoriamente temporais*, pois, mesmo que o tempo não seja explicitamente referenciado, o conceito temporal é implicitamente utilizado quando são analisados estados distintos de uma base de dados.

De acordo com a *formação de seus restringentes* (Figuras 2h), restrições podem ser *homogêneas* ou *heterogêneas*. Dentre as *homogêneas* estão as de: (i) *formação indefinida*, com restringentes de *tipo domínio*; (ii) *formação primitiva*, com restringentes compostos por objetos primitivos do modelo de dados adotado; e (iii) *formação construída*, com restringentes compostos por objetos construídos com a utilização de construtores do modelo, como o construtor de especialização. Quanto à *formação do restringido* (Figura 2.i), não é considerada a homogeneidade nem a *formação indefinida*.

Esses critérios dependem fortemente do esquema adotado pois, de acordo com a modelagem, alguns objetos da realidade podem ser representados através de objetos primitivos ou construídos [15].

Um dos principais componentes de uma restrição é sua condição restritiva, pois através dela se estabelece a relação entre restringido e restringente. Quanto à sua *formação*, condições restritivas são ditas *simples*, quando formadas por apenas uma comparação, ou *compostas*, quando formadas por duas ou mais comparações relacionadas por operadores lógicos. Condições *compostas* podem ainda ser *disjuntivas*, utilizando somente o operador *ou*, *conjuntivas*, utilizando somente o operador *e*, ou *múltiplas*, utilizando ambos operadores. Dessa forma, a Figura 2j ilustra essa classificação.

Quanto ao critério *completude* (Figura 2k), uma restrição é dita *completa*, quando representa inteiramente uma restrição da realidade modelada, ou *incompleta*, quando representa somente parte de uma restrição do mundo real, necessitando da existência de outras para representá-la. Restrições *incompletas* podem ser *geradoras*, quando necessitarem da geração de novas restrições *incompletas*, ou *geradas*, quando forem geradas a partir de outras restrições [14].

A representação de uma restrição real através de duas ou mais restrições no esquema facilita muito sua especificação e verificação. No entanto, elas devem ser ativadas e desativadas em conjunto e, possivelmente, possuir uma ordem pré-definida de verificação.

Restrições são ainda classificadas quanto a seu *propósito* (Figura 2l). Elas podem ter propósito: (i) *condicional* (caso em que são também conhecidas como *hard constraints*, *não violáveis*, ou *requeridas*), impedindo a efetivação da transação causadora da violação, quando violadas; (ii) *causativo*, estabelecendo ações executadas sobre a base após sua violação; ou (iii) *verificativo* (também conhecidas como *soft constraints*, *violáveis*, ou

*preferenciais*), especificando condições preferencialmente respeitadas, mas que não influenciam diretamente a execução de operações sobre a base.

Nota-se ainda uma relação íntima entre esse critério e o de *tratamento*, do aspecto *aplicação*, pois com base em seu *propósito*, serão definidas ações de violação e possíveis ações de prevenção de violação da restrição. Além disso, é obvio que restrições de *propósito causativo* devem possuir *pontos de verificação baseados em transação*.

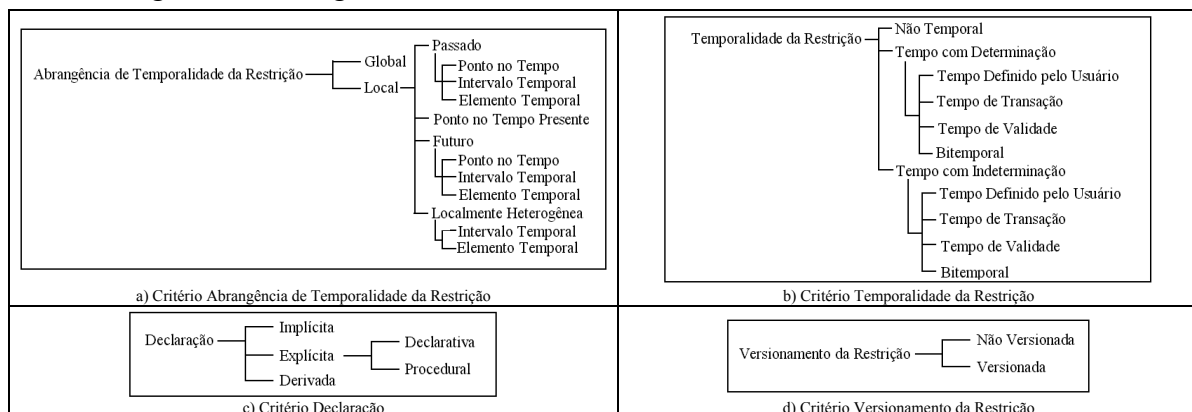
### 3.3. Especificação

O termo especificação de uma restrição de integridade diz respeito à sua incorporação ao esquema da base de dados [15]. Após isso, os dados deverão respeitá-la de forma incondicional.

O primeiro critério da *especificação* (Figura 3c) diz respeito à forma com que restrições são definidas sobre um BDTV. Assim, possuem *declaração implícita* as restrições inerentes ao modelo de dados, enquanto, restrições com *declaração explícita* são representadas explicitamente na especificação da base. De forma ideal, restrições *explícitas* deveriam corresponder somente a peculiaridades da realidade modelada, mas é comum sua utilização para suprir deficiências do modelo ou da modelagem utilizada. Além disso, elas podem ser *declarativas* ou *procedurais*.

Através do critério *temporalidade da restrição*, classificam-se restrições quanto à sua própria temporalidade e não quanto à temporalidade de seus restringentes e restringidos. Uma restrição é dita *não temporal* caso não tenha o conceito de tempo envolvido diretamente com ela. Restrições *temporais* possuem armazenados seus históricos de alterações, podendo ser válidas em determinados períodos. Elas possuem *tempo de validade*, *tempo de transação*, ambos tempos (*bitemporal*), ou *tempo definido pelo usuário*. Além disso, também estão relacionadas ao *tempo com determinação*, referindo-se a momentos com exatidão, ou ao *tempo com indeterminação*, referindo-se a momentos com pouca precisão [10]. A Figura 3b, ilustra essa classificação.

É importante citar que este critério é ortogonal ao aspecto *temporalidade*, pois o primeiro é inerente à restrição em si, enquanto o segundo diz respeito à temporalidade de seu restringente e restringido.



**Figura 3. Classificação quanto ao aspecto especificação.**

A *abrangência de temporalidade da restrição* pode ser *global*, quando a mesma for avaliada sobre todo o tempo de vida da base de dados, ou *local*, quando for avaliada sobre parte desse tempo. A abrangência *local* pode ainda envolver momentos: (i) no *passado*, através de um *ponto no tempo*, *intervalo temporal*, ou *elemento temporal*; (ii) no *presente*, através de um *ponto no tempo*; (iii) no *futuro*, através de um *ponto no tempo*, *intervalo temporal*, ou *elemento temporal*; ou (iv) em dois ou mais desses períodos (*localmente*

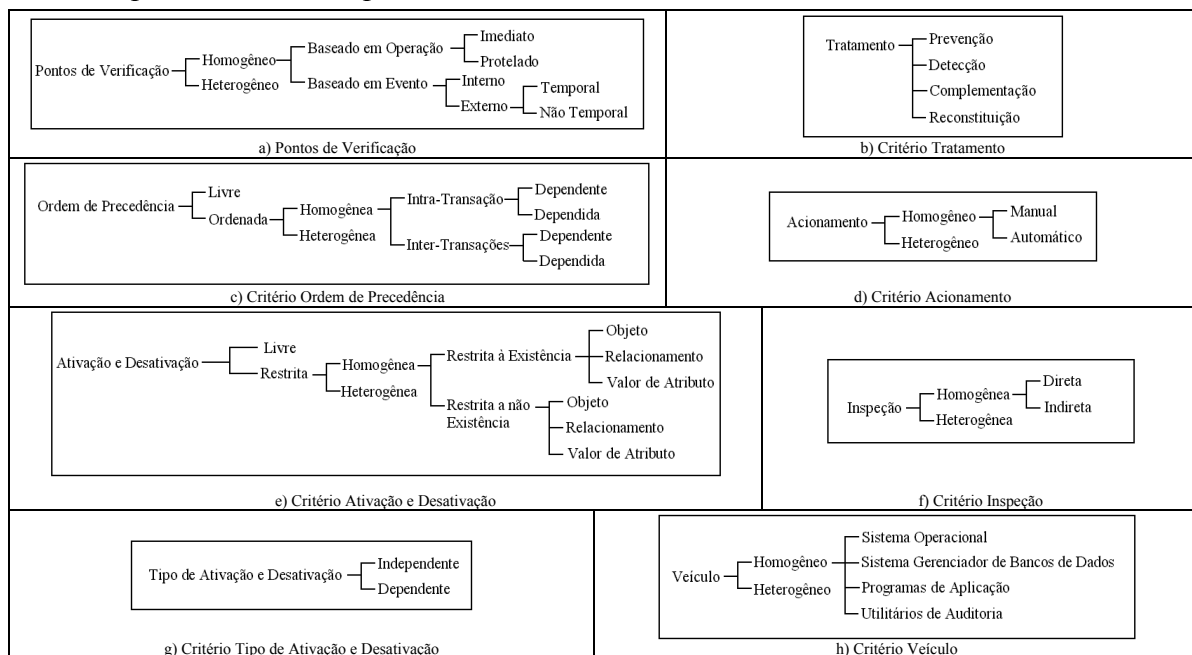
*heterogênea*), através de um *intervalo* ou *elemento temporal*. Restrições *não temporais* quanto ao critério anterior serão *globais* neste critério, por serem verificadas sobre todo o tempo de vida da base. A classificação pode ser vista na Figura 3a.

De forma análoga aos anteriores, está o critério *versionamento da restrição* (Figura 3d). Através dele, restrições são ditas *não versionadas* quando não possuem controle de versionamento sobre si ou *versionadas* caso contrário.

### 3.4. Aplicação

O aspecto aplicação desta classificação diz respeito à forma com que a integridade dos dados é mantida sobre as restrições existentes. Pode-se dizer que cada um dos critérios deste aspecto relaciona classes que representam alternativas de manutenção da integridade de uma base de acordo com as características de cada uma das restrições relacionadas a ela.

*Pontos de verificação* (Figura 4a) de restrições podem se basear em *operações* executadas sobre a base, sendo divididos em *imediatos*, que determinam momentos de verificação imediatamente posteriores à execução de cada operação, ou *protelados*, que determinam momentos de verificação posteriores à execução de transações completas [7]. Eles também podem se basear na ocorrência de *eventos internos* ou *externos* ao SGBD, sendo, muitas vezes, representados através de condições. Dentre os eventos considerados *internos* estão os relacionados à verificação de outras restrições, que são utilizados quando há *ordem de precedência* de verificação. Já os *eventos externos* podem estar relacionados a fatos *temporais* ou *não temporais*.



**Figura 4. Classificação quanto ao aspecto aplicação.**

De acordo com a Figura 4b, restrições também são classificadas sob o *tratamento* recebido por elas a fim de manter a integridade dos dados. Elas podem receber os seguintes *tratamentos*: (i) *prevenção*, que é baseado na execução de medidas que impeçam a violação; (ii) *deteção*, que se limita a acusar a violação necessitando de ações corretivas por parte de algum usuário da base; (iii) *complementação*, que consiste na execução de ações corretivas após a violação; ou (iv) *reconstituição*, que consiste na tomada de medidas, após a violação, que levem a base a um estado não influenciado pela operação ou



transação que violou a restrição. É importante notar que, normalmente, uma restrição é *tratada* de acordo com seu *propósito*.

Em relação à *ordem de precedência* (Figura 4c), uma restrição é considerada *ordenada*, quando existir alguma relação de ordem entre sua verificação e a verificação de outra(s), ou *livre*, caso contrário. Essa ordenação é dita *intra-transação*, quando estiver relacionada a uma transação única, ou *inter-transação*, quando a ordem tiver que ser obedecida por todas as verificações. Além disso, restrições *ordenadas* podem ser *dependentes*, quando sua verificação depender da verificação prévia de outra(s), ou *dependidas*, quando sua verificação for necessária para a verificação de outra(s) [6, 14]. É importante notar que a verificação de uma restrição diz respeito à análise de sua condição restritiva e também à execução de possíveis ações corretivas.

O critério *acionamento* (Figura 4d) classifica restrições de acordo com a forma com que se inicia seu processo de verificação. Assim, uma restrição possui *acionamento*: (i) *manual*, quando o início de sua verificação for determinado de forma explícita por algum usuário da base, através de um comando isolado ou em um programa; ou (ii) *automático*, quando sua verificação se iniciar devido ao controle exercido pelo SGBD, com base em seus pontos de verificação. Além disso, é possível que restrições se enquadrem em ambas classes, sendo ditas *heterogêneas*. Nota-se também que restrições com *acionamento homogêneo manual*, não possuem pontos de verificação pré-definidos.

Restrições de integridade podem estar ativas e exigirem que os dados as respeitem ou, inativas, quando, temporariamente, não estiverem influenciando a manutenção da integridade dos dados. Assim, a *ativação e desativação* (Figura 4e) de uma restrição pode ser: (i) *livre*, quando sua ocorrência for independente ou, (ii) *restrita*, quando sua ocorrência depender dos dados presentes na base. Essa segunda classe possui restrições *restritas à existência* ou a *não existência* de *objeto(s)*, *relacionamento(s)* ou *valor de atributo(s)*. Nota-se ainda que uma restrição com *ativação e desativação restrita* sempre pode ser representada por uma restrição *livre* com uma condição explícita de existência ou não existência. Assim, a condição restritiva representaria algo como: “se o objeto (não) existir, deverão ser verdadeiras as condições...”.

A fim de verificar uma restrição, são necessárias inspeções sobre seus restringentes e restringidos. Baseado nisso, o critério *inspeção* relaciona a possibilidade de inspeções *diretas* aos dados ou *indiretas*, quando metadados forem utilizados com o propósito de melhoria de performance. Nota-se ainda a necessidade de classificação da homogeneidade da inspeção, de acordo com a Figura 4f.

Quanto ao critério *tipo de ativação e desativação* (Figura 4g), restrições podem ser *dependentes*, com ativação e desativação relacionadas à(s) de outra(s), devendo ocorrer em conjunto, ou *independentes*, caso contrário [6]. Muitas vezes restrições *incompletas*, quanto ao critério *completude*, enquadram-se na primeira classe, por representarem em conjunto restrições reais.

O critério *veículo* (Figura 4h) da classificação é baseado no veículo utilizado para a verificação e manutenção da integridade da base. Dentre os veículos classificados estão o *sistema operacional*, o *sistema gerenciador de bancos de dados*, os *programas de aplicação* e os *utilitários de auditoria*. A utilização de qualquer um dos dois últimos veículos citados, a fim de manter a integridade dos dados, é devida à incapacidade ou à inadequação do SGBD às necessidades existentes [15].

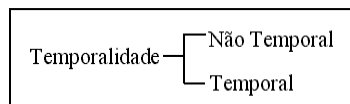
### 3.5. Temporalidade

Um dos principais aspectos desta classificação é o aspecto *temporalidade*, que classifica restrições de acordo com a temporalidade de seus restringidos e restringentes. A

temporalidade da própria restrição não será aqui considerada, por ter sido considerada na *aplicação*. Assim, de acordo com a Figura 5, classificam-se restrições *não temporais*, que envolvem restringidos e restringentes não temporais, e *temporais* que utilizam a temporalidade de forma implícita ou explícita.

A noção de tempo, aqui considerada, depende diretamente do *tempo real*, pois restrições *temporais* dependem da existência de um “relógio” cujo fluxo independe de qualquer ação efetuada sobre a base de dados. Com isso, sua violação, quando relacionada diretamente ao tempo, é irreversível devido à irreversibilidade do fluxo do tempo [4]. Devido a esse fato, essas restrições costumam possuir *propósito causativo*, sendo relacionadas a um conjunto de ações compensatórias, ou *verificativo*, para que sua violação não impeça a utilização da base.

Considerando os objetivos deste trabalho, restrições *temporais* são classificadas de acordo com diversos sub-critérios descritos a seguir. Deve-se citar que, apesar de haver, de acordo com esses sub-critérios, a possibilidade de existência de restrições cujos restringentes e restringidos possuam *tipo de tempo não temporal*, obviamente não será considerada essa possibilidade.



**Figura 5. Classificação quanto ao aspecto temporalidade.**

O sub-critério *granularidade temporal* permite distinguir restrições em relação à granularidade temporal de seus restringentes e restringidos. Refere-se aqui à unidade de medida utilizada em referências temporais, comumente ligada à unidade mínima temporal (*chronon*) representada pela base. Assim, classificam-se restrições com *granularidade homogênea* e *heterogênea* (Figura 6a). É importante salientar que, devido à sua simplicidade, classificou-se em conjunto a granularidade de restringentes e restringidos, apesar disso, se necessária, essa distinção pode ser facilmente efetuada.

Integrantes do restringente, provenientes da base de dados, podem possuir características temporais implícitas. Já os integrantes não provenientes da base podem estar, explicitamente, relacionando-a ao tempo. Assim, o *tipo de tempo do restringente* (Figura 6b) é classificado de forma análoga à *temporalidade da restrição*, exceto pela possibilidade de restringentes *heterogêneos*. Integrantes de restringentes *não temporais* serão de *tipo esquema*, *entidade* ou *relacionamento*, definidos como não temporais na modelagem de dados ou, de *tipo domínio* não relacionando explicitamente parte da base ao tempo. Os demais integrantes são *temporais*. A Figura 6c classifica o *tipo de tempo do restringido* de forma análoga, exceto pela impossibilidade de integrantes de tipo *domínio* e restringidos *heterogêneos*.

Referências, presentes em condições restritivas, relacionam restringidos e restringentes limitando a gama de valores possíveis do primeiro conjunto, com base no segundo. Baseando-se em [6, 10], conclui-se que essas referências podem utilizar duas noções distintas de tempo: (i) *tempo absoluto*, quando existe uma relação *quantitativa* entre restringido e restringente e, são especificados valores absolutos para a duração de evento(s) ou restringidas suas distâncias temporais; e (ii) *tempo relativo*, quando a relação é *qualitativa*, baseada em operadores especiais, como os operadores *antes*, *depois* e *durante*. O *tempo relativo* pode ainda estar relacionado ao *tempo atual (now)*, *tempos pré-definidos* ou *tempos de ocorrência de determinados eventos*, que podem ser *imprevisíveis*, *previsíveis periódicos* ou *previsíveis aperiódicos* [7, 9]. Além disso, conforme ilustrado na Figura 6d, foi classificada a homogeneidade e a possibilidade de existência de referências *não temporais*, para restrições com restringido ou restringente temporal, que não referenciem a temporalidade dos mesmos em suas condições restritivas.

Percebe-se que as *abrangências do restrigente e restringido*, do aspecto *substância*, não levam em consideração aspectos temporais, classificando a abrangência de apenas um estado da base de dados. Por isso, são necessários os sub-critérios de *abrangência temporal* para que, em conjunto com os critérios da *substância*, seja possível a análise da abrangência em relação a diversos estados da base ao longo de seu tempo de vida. As *abrangências temporais do restrigente e restringido* são classificadas de forma idêntica à *abrangência temporal da restrição*, presente no aspecto *especificação*, exceto por não estarem relacionadas à restrição e sim a seus restrigentes e restringidos. As Figuras 6e e 6f ilustram a classificação por ambos sub-critérios.

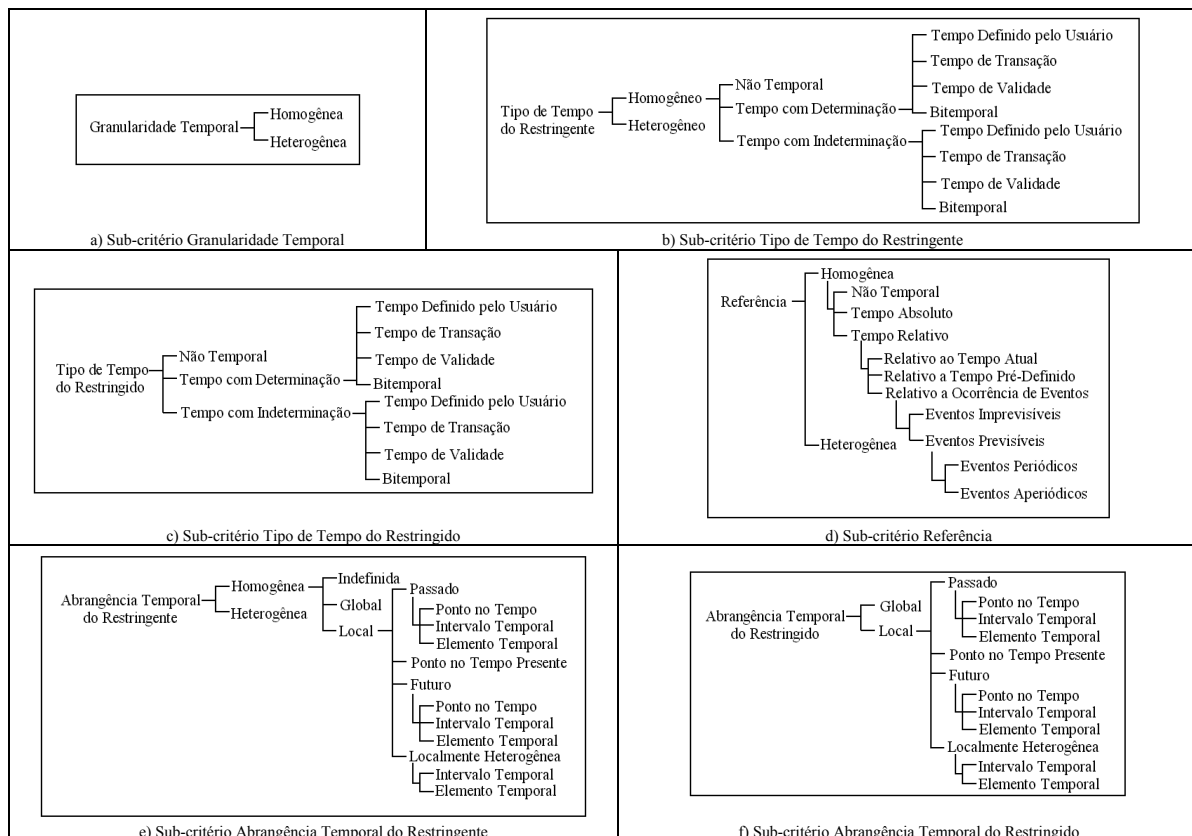
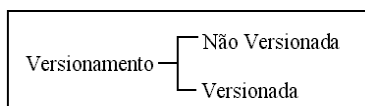


Figura 6. Classificação quanto ao critério temporal.

### 3.6. Versionamento

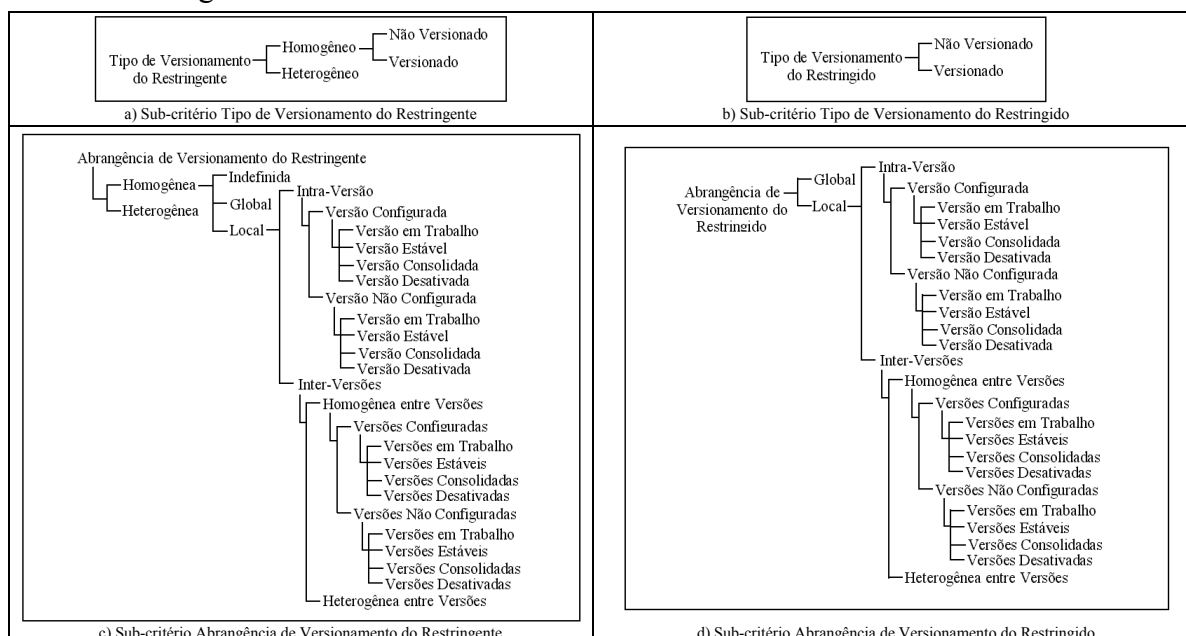
Além do conceito de versionamento na própria restrição, considerado na *especificação*, restrigentes e restringidos também podem utilizar esse conceito. Dessa forma, fez-se necessária a criação do aspecto *versionamento* que classifica restrições como *não versionadas*, quando seus restringidos e restrigentes não utilizam o conceito de versão, ou como *versionadas*, caso contrário. Restrições *versionadas* são ainda classificadas de acordo com diversos sub-critérios expostos durante esta seção. A Figura 7 ilustra o primeiro nível da classificação por esse aspecto.

Quanto ao *tipo de versionamento do restrigente* (Figura 8a), restrições são classificadas *versionadas* quando os integrantes de seus restrigentes forem de *tipo esquema*, *entidade* ou *relacionamento*, definidos como versionados na modelagem da base de dados ou, forem de *tipo domínio*, referenciando explicitamente o versionamento de alguma parte da base. Restrições que não se enquadram nessa classe são ditas *não versionadas*. Já o *tipo de versionamento do restringido* é classificado de forma análoga, não considerando a possibilidade de heterogeneidade e nem de restringidos de *tipo domínio*. Essa classificação pode ser vista na Figura 8b.



**Figura 7. Classificação quanto ao aspecto versionamento.**

De acordo com o versionamento, um restrigente pode possuir abrangência: (i) *indefinida*, com integrantes de tipo *domínio*, não diretamente relacionados ao versionamento da base; (ii) *global*, envolvendo todas as versões de componentes da base; ou (iii) *local*, envolvendo parte das versões de componentes. A abrangência *local* pode estar relacionada a uma única versão (*intra-versão*), que pode ser *configurada* ou *não configurada* e estará em um dos seguintes estados: *em trabalho*, *estável*, *consolidada* ou *desativada*. Além disso, essa abrangência também pode estar relacionada a duas ou mais versões distintas (*inter-versões*), as quais possuirão as características citadas acima. Considera-se ainda que restrigentes, *locais a duas ou mais versões*, podem possuir *abrangência de versionamento homogênea* ou *heterogênea entre versões*, conforme pode ser visto na Figura 8c.



**Figura 8. Classificação quanto ao critério versionada.**

A classificação e todas as considerações sobre a *abrangência do restrigente* também se aplicam à *abrangência do restringido*, exceto pela inexistência de restringidos com abrangência heterogênea ou *indefinida*. Dessa forma, a Figura 8d ilustra a classificação por esse sub-critério.

É importante notar que estes sub-critérios estão fortemente relacionados aos outros critérios e sub-critérios de abrangência citados anteriormente. Através dos critérios do aspecto *substância*, classificam-se restrições segundo a abrangência de um único estado da base de dados. A abrangência relacionada a diversos estados da base, ao longo do tempo, é considerada através dos sub-critérios do aspecto *temporalidade*. Já esses sub-critérios consideram a abrangência do restrigente em relação a diferentes versões de componentes da base, possibilitando assim a classificação completa da abrangência de restrições em um BDTV.

## 4. Estudo de Caso

Nesta seção serão apresentados exemplos de restrições sobre a base de dados de uma aplicação de controle de produção de software, especificadas em linguagem coloquial e classificadas de acordo com aspectos definidos anteriormente. Devido à limitação de espaço, serão analisados somente os aspectos *origem*, *temporalidade* e *versionamento*. A Figura 9a ilustra o diagrama de classes utilizado pela aplicação, baseada na granularidade de minutos, construído sob o Modelo Temporal de Versões (TVM) [12].

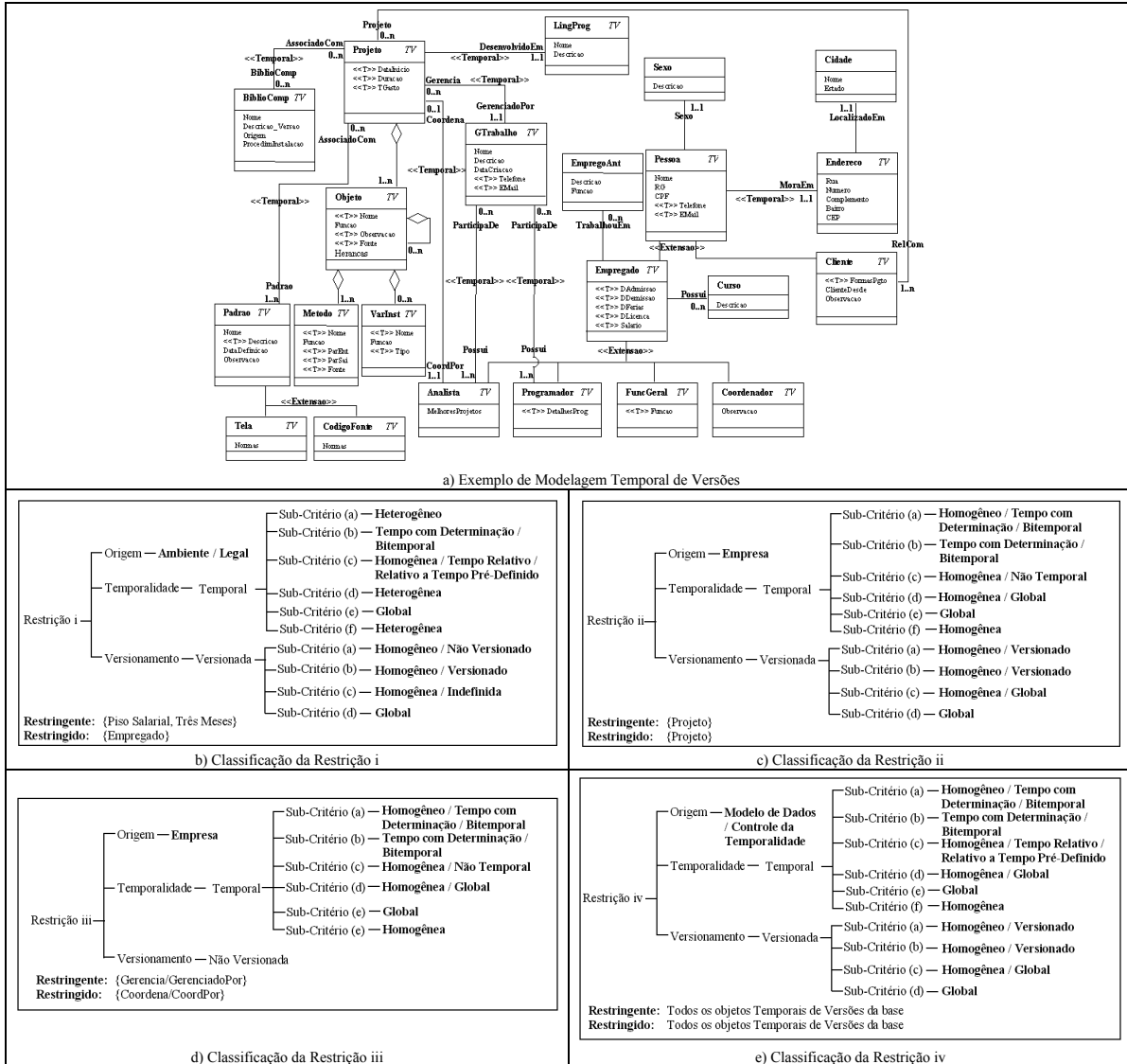


Figura 9. Classificação quanto ao critério versionada.

A aplicação exemplo possui a função de controlar processos de produção de software em uma empresa, desenvolvidos sob o paradigma da orientação a objetos. Projetos são compostos por módulos, representados por objetos, que por sua vez podem possuir outros objetos, métodos e variáveis de instância. Também são controlados os padrões de desenvolvimento e as bibliotecas utilizadas. Além disso, é gerenciado o quadro de funcionários e grupos de trabalho.

São classificadas as restrições: (i) o salário de um empregado não pode ser menor do que o piso salarial de sua categoria, após três meses de sua contratação; (ii) somente versões consolidadas de projetos terão gasto total definido; (iii) o analista coordenador de um projeto deverá participar do grupo de trabalho que o gerencia; e (iv) o tempo

*associado às versões de um objeto versionado deverá estar contido em seu tempo de vida.* Suas classificações são apresentadas nas Figuras 9b, 9c, 9d e 9e, através de árvores cujos nodos folha indicam as classes em que se enquadra cada restrição.

## 5. Trabalhos Relacionados

Estão presentes na literatura diversos trabalhos relacionados à classificação de restrições de integridade em bases de dados tradicionais [3, 14, 15]. Entretanto, são raras as classificações de restrições em um BDTV e, as poucas existentes não costumam classificá-las de forma explícita, além de tratarem apenas um dos conceitos envolvidos.

O trabalho [7] é a única exceção encontrada, pois nele, Doucet e outros classificam restrições de acordo com seu escopo (*scope*) e dependência (*binding*) sobre objetos, classes e versões. Ainda foram considerados a abrangência de estados, os pontos de verificação e as referências. Entretanto, características importantes não foram tratadas, como a temporalidade e o versionamento da própria restrição, sua abrangência temporal, tipos temporais e de versionamento, granularidade temporal, entre outros. Em [6], Dayal e outros apresentam uma boa classificação de restrições, considerando seu propósito, tipo de ativação e desativação, declaração, ordem de precedência, referência e abrangência do restringido e restringente, mas não consideram características de versionamento. Castilho, Casanova e Furtado [2] classificam restrições quanto à sua abrangência de estados, considerando também restrições temporais e de origem na implementação. Em seus respectivos trabalhos [1, 9], Böhlen e Escofet apresentam uma detalhada classificação, considerando o tipo de tempo de restringentes e restringidos, abrangência temporal e de estados, além da referência. Entretanto, o versionamento não foi considerado. Medeiros, Jomier e Cellary [11] apresentam uma boa classificação de restrições quanto à sua declaração, origem, abrangência de estados, aspecto e abrangência do restringente e restringido, considerando características de versionamento. Já Chomicki e Toman [4, 5] analisam a abrangência de estados, e o propósito de restrições temporais, considerando também características específicas de restrições de tempo real.

## 6. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma classificação detalhada de restrições de integridade em bases de dados temporais de versões. Restrições foram analisadas sob sua *origem, substância, especificação, aplicação, temporalidade e versionamento*, possibilitando a criação de classes ortogonais para cada aspecto, desde que não houvesse perda em abrangência. É importante deixar claro que esta classificação poderá ser útil para qualquer trabalho relacionado à manutenção eficiente e eficaz da integridade dos dados de bases temporais de versões, independentemente de outras características do modelo de dados ou do SGBD adotado. Além disso, devido à existência de aspectos específicos para a temporalidade e o versionamento, a classificação pode ser facilmente adaptada para bancos de dados tradicionais ou que apliquem somente um desses conceitos.

A classificação apresentada se diferencia de outras por priorizar e detalhar aspectos temporais e de versionamento, não deixando de lado os outros aspectos existentes. A fim de validar o trabalho, foi apresentado um estudo de caso classificando restrições, referentes a um exemplo de modelagem, de acordo com as classes definidas.

Acredita-se que o próximo passo, em busca da manutenção da integridade de dados em um BDTV, consiste na definição de formas para sua especificação e verificação. Além disso, a otimização desse processo é considerada essencial, a fim de possibilitar a utilização comercial desse tipo de base de dados. Com o intuito de alcançar esses objetivos,

experimentos e implementações de especificação e verificação de restrições estão sendo realizados, utilizando o modelo TVM [12] e sua linguagem de consulta TVQL [13].

## Referencias

1. BÖHLEN, Michael H. Valid Time Integrity Constraints. Tucson, AZ: Department of Computer Science – University of Arizona, 1994. (Technical Report 94-30)
2. CASTILHO, J. M. V. de; CASANOVA, M. A.; FURTADO, A. L. A Temporal Framework for Database Specifications. In: Intl. Conf. on Very Large Databases, VLDB, 8., 1982. Proceedings... Mexico City, Mexico:[s.n.], 1982. p. 280-291.
3. CASTILHO, José Mauro V. Especificações Formais e Sistemas de Bancos de Dados. Buenos Aires: Kapelusz, 1987. 157p.
4. CHOMICKI, Jan. Real-Time Integrity Constraints. In: Intl. Symposium on Principles of Database Systems, ACM SIGACT-SIGMOD-SIGART, 11., 1992. Proceedings... San Diego, California, United States:[s.n.], 1992. p. 274-282.
5. CHOMICKI, Jan; TOMAN, David. Implementing Temporal Integrity Constraints Using an Active DBMS. IEEE Transactions on Knowledge and Data Engineering, [S.l.], v.7, n.4, p. 566-582, Feb. 1995.
6. DAYAL, Umeshwar. et al. The HiPAC Project: Combining Active Databases and Timing Constraints. ACM SIGMOD Record, [S.l.], v.17, n.1, p. 51-70, Mar. 1988.
7. DOUCET, Anne et al. Using Database Versions to Implement Temporal Integrity Constraints. In: Constraint Database and Applications, CDB, 2., 1997. Proceedings... Delphi, Greece:[s.n.], 1997. p. 219-233.
8. EDELWEISS, Nina; OLIVEIRA, José Palazzo Moreira de. Modelagem de Aspectos Temporais de Sistemas de Informação. Recife: UFPE-DI, 1994. 162p.
9. ESCOFET, Carme Martín. Analyzing Temporal Integrity Constraints to Obtain the Minimum Number of Transition Rules. Barcelona, Catalunya: Universitat Politècnica de Catalunya, 2001. (Technical Report LSI-01-52-R)
10. JENSEN, C.S. et al. The Consensus Glossary of Temporal Database Concepts. In: ETZION, O.; JAJODIA, S.; SRIPADA, S. (eds.) Temporal Databases Research and Practice. Berlin-Heidelberg: Springer-Verlag, 1998. p. 367-405.
11. MEDEIROS, Claudia Bauzer; JOMIER, Geneviève; CELLARY, Wojciech. Maintaining Integrity Constraints across Versions in a Database. Campinas, SP: DCC–Unicamp, 1992. (Relatório Técnico DCC-08/92)
12. MORO, M. M.; SAGGIONARO, S. M.; EDELWEISS, N.; SANTOS, C. S. Adding time to an object oriented versions model. In: Intl. Conf. on Database and Expert Systems Applications, DEXA, 12., 2001. Proceedings... Munich, Germany: v. 2113 of LNCS, p. 805-814, Sep. 2001.
13. MORO, M. M.; EDELWEISS, N.; ZAUPA, A.; SANTOS, C. S. TVQL - Temporal Versioned Query Language. In: Intl. Conf. on Database and Expert Systems Applications, DEXA, 13., 2002. Proceedings... Aix-en-Provence, France: v. 2453 of LNCS, p. 618-627, Sep. 2001.
14. RAM, D. Janaki et al. Constraint Meta-Object: A New Object Model for Distributed Collaborative Designing. IEEE Transactions on Systems, Man, and Cybernetics, [S.l.], v.27, n.2, p. 208-221, Mar. 1997.
15. SANTOS, Clesio Saraiva dos. Caracterização Sistemática de Restrições de Integridade em Bancos de Dados. 1980. Tese de Doutorado – D.I-PUC, Rio de Janeiro.